In the Specification:

Please substitute the following paragraphs for the corresponding paragraphs beginning at the indicated location in the specification as originally filed (some of which have been previously amended).

Page 1, line 12+:

When a diagnosis system finds a product of a semiconductor device defective, the fault analyzing system traces the signal propagation paths from the output signal pins toward the upstream, and nominates the candidates of a fault origin which are to satisfy the output signal pattern at the output signal nodes. Examples of the fault analyzing system are disclosed in Japanese Patent Publication of Unexamined Application Nos. 8-146093 and 10-062494. Japanese Patent Publication of Unexamined Application Nos. 8-146093 and 10-062494 have resulted in Japanese Patent Nos. 2655105 and 2921502, respectively. These prior art fault analyzing systems firstly extract partial circuits related to the output terminals where the output signal pattern ~~different~~ differs from an expected signal pattern, and presume the signal propagation paths of the output signal pattern. If necessary, the prior art fault analyzing systems further extract partial circuits closer to the signal input terminals, and nominate the candidates of the signal propagation paths from the fault origin. The prior art fault analyzing systems weight the nodes on the candidates on the basis of the connections in the candidates, and determine the heaviest nodes. The prior art fault analyzing systems output the pieces of analysis information representative of the fault origins.

Page 9, Line 4+:

It is therefore an important object of the present invention to provide a fault analyzing system, which is

available for various kinds of fault such as, for example, the open fault, the bridge fault and the ~~stuck- up~~ stuck-at fault without consuming a long time period.

Page 11, line 5+:

Fig. 5 is a block diagram showing the arrangement of a ~~stuck- up~~ stuck-at fault weighting means together with a data memory unit and a fault candidate memory unit;

Page 12, line 8+:

Fig. 11 is a flowchart showing a sequence of jobs executed by the ~~stuck- up~~ stuck-at fault weighting means;

Page 15, line 15+:

The fault candidate memory unit 5 includes a ~~stuck- up~~ stuck-at fault candidate memory 51, an open-fault candidate memory 52, a bridge- fault candidate memory 53 and a mixed fault candidate memory 54. Pieces of ~~stuck- up~~ stuck-at fault candidate information are stored in the ~~stuck- up~~ stuck-at candidate memory 51, and are representative of candidates of ~~stuck- up~~ stuck-at fault and fault terminals related to the ~~stuck- up~~ stuck-at fault candidates. Pieces of open- fault candidate information are stored in the open- fault candidate memory 52, and are representative of candidates of open- fault and fault terminals related to the open-fault candidates. Pieces of bridge- fault candidate information are stored in the bridge- fault candidate memory 53, and are representative of candidates of bridge- fault and fault terminals related to the bridge- fault candidates. Pieces mixed fault candidate information are stored in the mixed fault candidate memory 54, and are representative of candidates of

mixed fault and fault terminals related to the ~~stuck-up~~ stuck-at fault candidates.

Page 16, line 6+:

The fault propagation path presuming unit 2 includes a fault terminal searching means 21, a partial circuit extracting means 22, an internal logic state presuming means 23, a logic state registration means 24 and a first related terminal registration means 25. The fault terminal searching means 21 accesses the pieces of data information stored in the circuit configuration memory 41 and the logic state memory 42, and looks for a failure output terminal or terminals. If the fault terminal searching means 21 does not find any fault terminal, the fault propagation path presuming unit 2 ~~terminals~~ terminates the presuming operation. The partial circuit extracting means 22 accesses the pieces of data information stored in the circuit configuration memory 41, and determines partial circuits related to the fault terminals already found by the fault terminal searching means 21. The internal logic state presuming means 23 accesses the pieces of data information stored in the circuit configuration memory 41 and the logic state memory 42, and presumes fault propagation paths on the basis of the layout of the partial circuits and the logic state at the boundaries of the partial circuits. The logic state registration means 24 registers the logic state presumed by the internal logic state presuming means 23 in the logic state memory 42. The first related terminal registration means 25 investigates the fault propagation paths presumed by the internal logic state presuming means 23, and determines the nodes on the fault propagation paths so as to write pieces of related fault terminal information representative of the fault terminals related to the nodes on the fault propagation paths in the first related fault terminal

memory 43.

Page 17, line 6+:

The fault candidate weighting unit 3 includes a ~~stuck- up~~ stuck-at fault weighting means 31, an open-fault weighting means 32, a bridge- fault weighting means 33 and a mixed fault candidate output means 34. The ~~stuck- up~~ stuck-at fault is a kind of fault, which causes a node to be in a certain logic level at all times. Even if a doubtful node changes the logic state with time, the doubtful node is a candidate of the ~~stuck- up~~ stuck-at fault in so far as the fault state is not propagated to the output terminal. The ~~stuck- up~~ stuck-at fault weighting means 31 gives a weight to candidates of the ~~stuck- up~~ stuck-at fault. Of course, the ~~stuck- up~~ stuck-at weighting means 31 gives the weight to the doubtful node.

Page 18, line 1:

Turning to figure 5 of the drawings, the ~~stuck- up~~ stuck-at fault weighting means 31 includes a related fault terminal information searching means 311, a first related fault terminal information merging means 312, a merge judging means 313, a candidate sorting means 314 and a candidate outputting means 315. The related fault terminal information searching means 311 searches the first related fault terminal memory 43 for the pieces of related fault terminal information representative of the fault terminals of nodes on the fault propagation paths at a given time.

Page 18, line 14+:

The merge judging means 313 monitors the merging work to see whether or not the first related fault terminal information merging means 312 completes the merging work for all the times. The candidate sorting means 314 is operative to sort the candidates of the

~~stuck- up~~ stuck-at fault with the number of related fault terminals, which is determined by the first related fault terminal information merging means 312, as the weight. The candidate sorting means 314 forms a list of the stuck-at fault. The candidate outputting means 315 delivers the pieces of stuck-up fault candidate information representative of the list and the related fault terminals to the ~~stuck- up~~ stuck-at fault candidate memory 51 and the output unit 6.

Page 20, line 1+:

~~Turing~~ Turning to figure 7 of the drawings, the bridge- fault weighting means 33 includes the related fault terminal information searching means 321, the second related fault terminal information merging means 322, a candidate selecting means 331, an unqualified candidate removing means 332, a third related fault terminal information merging means 333, a merge judging means 334, a candidate sorting means 335 and a candidate outputting means 336. The related fault terminal information searching means 321 and the second related fault terminal information merging means 322 have been already described in connection with the open- fault weighting means 32.

Page 21, line 10+:

Turning to figure 8 of the drawings, the mixed fault candidate output means 34 includes a candidate memory searching means 341, a double candidate eliminating means 342, a mixed fault candidate sorting means 343 and a mixed fault candidate outputting means 344. The candidate memory searching means 341 reads out the candidates of the ~~stuck- up~~ stuck-at fault, the candidates of the open- fault and the candidates of the bridge- fault from the ~~stuck- up~~ stuck-at fault candidate memory 51, the open- fault candidate memory 52, the bridge- fault candidate memory 53 and the mixed

fault candidate memory 54, and draws up a list of mixed fault candidates.

Page 21, line 19+:

The double candidate eliminating means 342 checks the list of the mixed fault candidates to see whether each node is listed as both of the ~~stuck- up~~ stuck-at fault candidate and the open- fault candidate with the pieces of related fault terminal information identical with one another. If a node is listed as both of the ~~stuck- up~~ stuck-at fault candidate and if the open- fault candidate and the pieces of related fault terminal information for the ~~stuck- up~~ stuck-at fault candidate are identical with the pieces of related fault terminal information for the open- fault candidate, the double candidate eliminating means 342 eliminates the open- fault candidate from the list of mixed fault candidates. After the elimination of the open- fault candidates from the list, the double candidate eliminating means 342 transfers the list of mixed fault candidates to the mixed fault candidate sorting means 343.

Page 22, line 15+:

The fault analyzing system according to the present invention achieves the fault analysis through a sequence shown in figure 9. An integrated circuit is assumed to be analyzed. The fault analyzing system firstly renders the fault propagation path presuming unit 2 presuming fault propagation paths from the failure output terminals toward the upstream as by step A1. Subsequently, the fault analyzing system renders the ~~stuck- up~~ stuck-at fault weighting means 31,the open- fault weighting means 32 and the bridge- fault weighting means 33 determining candidates of the ~~stuck- up~~ stuck-at fault through the weighting as by step A2, candidates of the open- fault through the weighting as

by step A3 and candidates of the bridge- fault through the weighting as by step A4. Finally, the fault analyzing system renders the mixed fault candidate output means 34 delivering the pieces of mixed fault candidate information representative of the list of the mixed fault candidates as by step A5.

Page 24, line 22+:

The job at step A2 is hereinbelow described in detail with reference to figures 5 and 11. In the following description, a0 and a1 represents a ~~stuck-up~~ stuck-at fault fixed to logic level and a ~~stuck-up~~ stuck-at fault fixed to logic level, respectively.

Page 25, line 3+:

The related fault terminal information searching means 311 accesses the pieces of related fault terminal information representative of the fault terminals, and determines pieces of related fault terminal information representative of the fault terminals ~~relate~~ related to the nodes on the fault propagation path or paths at a given time and the next time as by step B1. The first related fault terminal information merging means 312 merges the pieces of related fault terminal information at each node as by step B2 in so far as the logic state is unchanged. Node A is assumed to be on the failure propagation path at time T2 as well as time T1 without changing the logic state such as logic level. If the expected logic state is logic level, the fault terminal related to the ~~stuck-up~~ stuck-at candidate A-sa0 is presumed by merging the piece of related fault terminal information at time T1 with the piece of related fault terminal information at time T2. However, ~~it~~ if the logic state is different between time T1 and time T2, the first related fault terminal information merging means 312 does not merge those

pieces of related fault terminal information, because the pieces of related fault terminal information are assumed to represent different faults A-sa0 and A-sa1.

Page 26, line 3+:

When the pieces of related fault terminal information representative of the fault terminals at the latest time are processed at step B2, the answer at step B3 is given negative, and the candidate sorting means 314 sorts the ~~stuck- up~~ stuck-at fault candidates with the number of related fault terminals used as the weight as by step B4. As a result, the candidate sorting means 314 draws up a list of ~~stuck- up~~ stuck-at fault candidates. Finally, the candidate outputting means 315 delivers the pieces of ~~stuck- up~~ stuck-at fault candidate information representative of the list of ~~stuck- up~~ stuck-at fault candidates and the related fault terminals to the output unit 6, and stores them in the ~~stuck- up~~ stuck-at fault candidate memory 51 as pieces of ~~stuck- up~~ stuck-at fault candidate information.

Page 27, line 11+:

When the pieces of related fault terminal information representative of the fault terminals at the latest time are processed at step C2, the answer at step C3 is given negative, and the candidate sorting means 324 sorts the open- fault candidates with the number of related fault terminals used as the weight as by step C4. As a result, the candidate sorting means 324 draws up a list of open- fault candidates. Finally, the candidate outputting means 325 delivers the pieces of open- fault candidate information representative of the list of open- fault candidates and the related fault terminals to the output unit 6, and stores them in the open- fault candidate memory 52 as pieces of open- fault candidate information.

The job at step A4 is hereinbelow described in detail with reference to figures 7 and 13. The related fault terminal information searching means 321 searches the first related fault terminal memory 43 for pieces of related fault terminal information representative of the fault terminals related to the nodes on the failure propagation path or paths at a given time and the next time as by step C1, and the second related fault terminal information merging means 322 merges the pieces of related fault terminal information at the given time and the pieces of related fault terminal information at the next time as by step C2. Thus, the jobs at the first two steps are similar to those in the flowchart shown in figure 12.

Page 28, line 8+:

Subsequently, candidate selecting means 331 selects two candidates from the open- fault candidates listed at step C2, and forms a candidate pair as by step D1. The unqualified candidate removing means 332 checks the candidate pair to see whether or not it meets conditions of a bridge- fault candidate, and removes the candidate pair in so far as it does not meet the conditions as by step D2. If the expected logic state is same at a time at which the fault takes place, the candidate pair is not any origin of the bridge- fault, and the candidate pair is removed. If the candidates of the pair does not cross in the layout of the circuit layout, the candidate pair is not any origin of the bridge- fault. Even if the pieces of related fault terminal information are merged with one another, the number of fault terminals may be small. The candidate pair is rarely an origin of the bridge fault, and the candidate pair is removed. Moreover, if one of the candidates of the pair contains all the fault terminals, the candidate is possibly listed as the ~~stuck- up~~ stuck-at fault candidate or the open-

fault candidate, and it is removed. As a result, the bridge- fault weighting means is released from unnecessary jobs described hereinbelow. Thus, only candidate pair at a high possibility remains as a candidate of bridge- fault.

Page 30, Line 7+:

The job at step A5 is hereinbelow described in detail with reference to figures 8 and 14. The candidate memory searching means 341 reads out the pieces of ~~stuck- up~~ stuck-at fault candidate information representative of the list of ~~stuck- up~~ stuck-at candidates and the related fault terminals, the pieces of open- fault candidate information representative of the list of open- fault candidates and the related fault terminals and the pieces of bridge- fault candidate information representative of the list of bridge- fault candidates and the related failure terminals from the ~~stuck- up~~ stuck-at fault candidate memory 51, the open- fault candidate memory 52 and the bridge- fault candidate memory 53 as by step E1, and draws a list of mixed fault candidates. The double candidate eliminating means 342 checks the list of mixed fault candidates to see if there are an open- fault candidate and a ~~stuck- up~~ stuck-at fault candidate both relating to a node and having the pieces of related fault terminal information shared therebetween, and eliminates the open- fault candidate from the list of mixed fault candidates with the positive answer as by step E2.

Page 30, Line 22+:

As described hereinbefore, the difference between the ~~stuck-up~~ stuck-at fault candidate and the open- fault candidate is whether the logic state is to be taken into account or not. In other words, the pieces of related fault terminal information representative of

the open- fault candidate A-open is equivalent to the pieces of related fault terminal information representative of the stuck-up fault candidates A-sa0 and A-sa1 ORed with each other. In case where only the stuck-at fault candidate A-sa0 is found in the list of stuck-at candidates, the stuck-at fault candidate A-sa0 is substantially identical with the open- fault candidate A- open, and the same pieces of related fault terminal information are shared between the stuck-at fault candidate A-sa0 and the open- fault candidate A-open. In this situation, if both of the stuck-at fault candidate A-sa0 and the open- fault candidate A- open are written into the list of mixed fault candidates, the piece of stuck-at fault candidate information is doubled with the piece of open- fault candidate information. In order to reduce the pieces of mixed fault candidate information, the double candidate eliminating means 342 eliminates the pieces of open-fault candidate information from the list of mixed fault candidates.

Page 33, line 3+:

The first related terminal registration means 25 stores pieces of related fault terminal memory 43 at step A16. In detail, the ~~firsts~~ first related fault terminal registration means 24 stores a piece of related fault terminal information representative of the fault terminal F1 related to the node on the fault propagation path p12 in the partial circuit c1, pieces of related fault terminal information representative of the fault terminals F2 and F3 related to the node on the propagation path p13 in the partial circuit c2, a piece of related fault terminal information representative of the fault terminal F2 related to the node on the propagation path p14 in the partial circuit c2, a piece of related fault terminal information representative of the fault terminal F3 related to the

node on the propagation path p15 in the partial circuit c2 and a piece of related fault terminal information representative of the fault terminals F4 related to the node on the propagation path p16 in the partial circuit c3 in the first related fault terminal memory 43.

Page 35, line 6+:

Nodes A, B, C and D are assumed to be found on failure propagation paths in time planes T1 and T2 through the presumption as shown in figures 16, 17 and 18. Figures 17 and 18 show the time planes concurrently produced. The expected logic state at these nodes A, B, C and D is placed in the brackets. The ~~stuck- up~~ stuck-at fault weighting means 31 firstly gives weights to the nodes A, B, C and D as follows. The related fault terminal information searching means 311 searches the first related fault terminal memory 43 for the fault terminals on the fault propagation paths at step B1 (see figure 11). The related fault terminal information searching means 311 presumes that a ~~stuck-up~~ stuck-at fault is propagated from the ~~stuck- up~~ stuck-at fault candidate A-sa0 to F1, F2, F3 and F4, from the ~~stuck- up~~ stuck-at fault candidate B-sa0 to F1, F2, F3 and F4 and from the ~~stuck- up~~ stuck-at fault candidate C-sa1 to F1 and F2 at time T1 as shown in figure 16. The related fault terminal information searching means 311 further presumes that the ~~stuck- up~~ stuck-at fault candidates A-sa1, B-sa0 and C-sa1 respectively relate to F5/ F6, F5/ F6 and F5 at time T2 as shown in figure 17.

Page 35, Line 21+:

The first related fault terminal information merging means 312 merges the pieces of related fault terminal information representative of the nodes at time T1 with the pieces of related fault terminal information representative of the nodes at time T2 at

step B2 on the condition that the logic state is identical between time T1 and time T2. Only the ~~stuck-up~~ stuck-at fault candidates A and B keep the logic state. As a result, the first related fault terminal information merging means 312 obtains pieces of related fault terminal information representative of F1, F2, F3, F4, F5 and F6 related to the node B and pieces of related fault terminal information representative of F1, F2 and F5 related to the node C.

Page 36, Line 8+:

The merge judging means 313 investigates whether or not the first related fault terminal merging means 312 completes the job on all the time planes, and finds the time plane at time T2 shown in figure 18 not to be processed. The ~~stuck-up~~ stuck-at fault weighting means 31 returns to step B1. The related fault terminal information searching means 311 searches the first related fault terminal memory 43 for the pieces of related fault terminal information at time T2, and reads out the pieces of related fault terminal information representative of F5 and F6 related to the node D-sa0. The first related fault terminal information merging means 312 further merges the pieces of related fault terminal information at step B2. The first related fault terminal information merging means 312 presumes that the ~~stuck-up~~ stuck-at fault candidates A-sa0, B-sa0, C-sa1, A-sa1 and D-sa0 relate to F1/ F2/ F3/ F4, F1/ F2/ F3/ F4/ F5/ F6, F1/ F2/ F5, F5/ F6 and F5/ F6, respectively.

Page 36, line 21+:

The merge judging means 313 finds that there remains no time plane at step B3, and the candidate sorting means 314 sorts the ~~stuck-up~~ stuck-at candidates with the number of related fault terminals used as the weight at step B4. The ~~stuck-up~~ stuck-at

fault candidate B-sa0 relating to F1, F2, F3, F4, F5 and F6, the ~~stuck- up~~ stuck-at fault candidate A-sa0 relating to F1, F2, F3 and F4, the ~~stuck- up~~ stuck-at fault candidate C-sa1 relating to F1, F2 and F5, the ~~stuck- up~~ stuck-at fault candidate A-sa1 relating to F5 and F6 and the ~~stuck- up~~ stuck-at fault candidate D-sa0 relating to F5 and F6 are listed in this order.

Page 37, line 10+:

Subsequently, the job at step A3 (see figure 9) is detailed with reference to figures 16 to 18. The related fault terminal information searching means 321 searches the first related fault terminal memory 43 for nodes on the fault propagation paths at step C1. An open- fault is possibly propagated from the open- fault candidate A- open to F1, F2, F3 and F4, from the open-fault candidate B-open to F1, F2, F3 and F4 and from the open- fault candidate C-open to F1 and F2 in the time plane T1 shown in figure 16. On the other hand, the open- fault is possibly propagated from the open-fault candidate A-open to F5 and F6, from the open-fault candidate B-open to F5 and F6 and from the open-fault candidate C-open to F5 in the time plane T2 shown in figure 17.

Page 39, line 22+:

Subsequently, the candidate selecting means 331 selects two open-fault candidates from the candidate list at step D1. As a result, three candidate pairs AB, AC and BC are obtained. The unqualified candidate eliminating means 332 examines the candidate pairs AB, AC and BC to see whether or not they meet the conditions of the bridge- fault candidate. If the expected logic state of one of the candidates is identical with the expected logic state of the other candidate at a time at which the fault is to take place, the unqualified candidate removing means 332

eliminates the candidate pair from the list of bridge-fault candidates. The candidates of the pair AB have the expected logic state [1] on the time plane T1, and the candidates of the pair AC have the expected logic state [0] on the time plane T2. For this reason, the unqualified candidate eliminating means 332 eliminates the candidate pairs AB and AC from the time planes T1 and T2. If one of the candidates of a pair has the related fault terminals perfectly contained by the other candidate of the same pair, the candidate has been already listed in the list of ~~stuck- up~~ stuck-at fault candidates or the list of open- fault candidates, and, for this reason, the unqualified candidate eliminating means 332 eliminates the candidate pair. The fault terminals related to the candidate C-open are perfectly contained in the fault terminal group of the candidate A- open and the fault terminal group of the candidate B-open. For this reason, the unqualified candidate eliminating means 332 eliminates the candidate pairs AC and BC from the list of bridge-fault candidates. As a result, all the candidate pairs are eliminated from the list of bridge- fault candidate.

Page 43, line 18+:

Subsequently, the double candidate eliminating means 342 checks the list to see whether or not any mixed fault candidate is doubled therein, and eliminates open- fault candidates same as ~~stuck- up~~ stuck-at fault candidates and having the fault terminals identical with those of the ~~stuck- up~~ stuck-at fault candidates from the list at step E2. The candidates B-sa0 and B-open stand in the list, and the fault terminals related to B-sa0 are identical with the fault terminals related to B-open. For this reason, the candidate B-open is eliminated from the list. Similarly, the double candidate eliminating means 342

eliminates the candidates C-open and D-open from the list. As a result, list of mixed fault candidates contains B-sa0 relating to the fault terminals F1, F2, F3, F4, F5 and F6, A-sa0 relating to the fault terminals F1, F2, F3 and F4, C-sa1 relating to the fault terminals F1, F2 and F5, A-sa1 relating to the fault terminals F5 and F6, D-sa0 relating to the fault terminals F5 and F6, A-open relating to the fault terminals F1, F2, F3, F4, F5 and F6 and AD-bf relating to the fault terminals F1, F2, F3, F4, F5 and F6.

Page 45, line 3+:

As will be understood from the foregoing description, the fault analyzing system according to the present invention presumes the open- fault candidates and the bridge- fault candidates as well as the ~~stuck- up~~ stuck-at fault candidates. Moreover, the fault analyzing system according to the present invention completes the analysis without consuming a long time, because the fault analyzing system presumes the fault candidates through merging the nodes of fault propagation paths on different time planes.

Page 51, line 20+:

Subsequently, the fault candidate weighting unit 3 renders the ~~stuck- up~~ stuck-at fault weighting means 31, the open- fault weighting means 32 and the bridge- fault weighting means 33 drawing up a list of ~~stuck- up~~ stuck-at fault candidates, a list of open- fault candidates and a list of bridge- fault candidates on the basis of the pieces of related fault terminal information on plural time planes as well as outputting the pieces of ~~stuck- up~~ stuck-at fault candidate information, the pieces of open- fault candidate information and the pieces of bridge- fault candidate information to the output unit 6 and the respective fault candidate memories 51, 52 and 53. The fault

candidate weighting unit 3 further renders the mixed fault candidate output means 34 drawing up a list of mixed fault candidates on the basis of the pieces of fault candidate information read out from the fault candidate memories 51, 52 and 53, and, thereafter, the mixed fault candidate output means 34 delivers the pieces of mixed fault candidate information to the output unit 6 and the mixed fault candidate memory 54. Thus, the fault propagation path presuming program and the fault candidate weighting program run on the data processor of the fault propagation path presuming unit 2/ 2A and the data processor of the fault candidate weighting unit 3 respectively, and achieves the jobs described in connection with the first and second embodiments.

Page 52, line 17+:

As will be appreciated from the foregoing description, the fault analyzing system according to the present invention searches the pieces of related fault terminal information for nodes identical in logic state between different time planes, and merges the pieces of related fault terminal information representative of the nodes identical in logic state between the different time planes for drawings up a list of weighted ~~stuck- up~~ stuck-at fault candidates. The fault analyzing system according to the present invention further searches the pieces of related fault terminal information for nodes relating to the fault terminals on the time planes, and merges the pieces of related fault terminal information on the different time planes for drawing up a list of weighted open-fault candidates. The fault analyzing system according to the present invention forms every two open- fault candidates into candidate pairs on each time plane, and merges the pieces of related fault terminal information for the candidates of the pairs for drawing up a list

of weighted bridge- fault candidates. Thus, the fault analyzing system according to the present invention merges the pieces of related fault terminal information on the plural time planes under the different conditions so as to ~~drawing~~ draw up the lists for the plural kinds of fault.